CLAIMS:

1. A method for switching VOIP packets in a data network, said method comprising the steps of:

receiving a first packet in a network switch;

determining if the first packet is a VOIP packet;

determining a dynamically negotiated VOIP port for a VOIP session from at least one of the first packet and a second packet received in the network switch, if the first packet is determined to be the VOIP packet; and

classifying all subsequent VOIP packets corresponding to the dynamically negotiated VOIP port in accordance with predetermined parameters.

2. A method for switching VOIP packets in a data network as recited in claim 1, wherein said steps of determining if the first packet is a VOIP packet, determining a dynamically negotiated VOIP port, and classifying subsequent VOIP packets are performed in a filtering step by a fast filtering processor.

3. A method for switching VOIP packets in a data network as recited in claim 2, wherein said filtering step further comprises:

applying a filter mask to a header of a packet;

extracting unmasked information;

comparing the unmasked information to a filtering table; and

executing predetermined filtering actions based upon the comparison to the filtering table.

4. A method for switching VOIP packets in a data network as recited in claim 1, wherein the step of determining if the first packet is a VOIP packet further comprises the steps of:

snooping a packet header of the first packet; and

determining if a VOIP well known port is contained in the packet header.

5. A method for switching VOIP packets in a data network as recited in claim 4, wherein said snooping step further comprises:

applying a filter mask to the packet header; and

comparing unmasked information from the header to entries in a filter table to determine a match.

6. A method for switching VOIP packets in a data network as recited in claim 5, wherein said step of determining if a VOIP well known port is contained in the packet header further comprises storing the well known port in the filtering table

upon initialization of the network switch.

7. A method for switching VOIP packets in a data network as recited in claim 1, wherein the step of determining a dynamically negotiated VOIP port further comprises determining a layer four port negotiated by at least two VOIP users for exclusive use in transmitting VOIP frames for a particular VOIP session.

8. A method for switching VOIP packets in a data network as recited in claim 1, wherein the step of determining a dynamically negotiated VOIP port further comprises at least one of extracting the dynamically negotiated VOIP port from the first packet and sending the second packet to a CPU for decoding and extraction of the dynamically negotiated VOIP port.

9. A method for switching VOIP packets in a data network as recited in claim 1, wherein the step of determining a dynamically negotiated VOIP port further comprises determining an RTP protocol port.

10. A method for switching VOIP packets in a data network as recited in claim 1, wherein the step of classifying all subsequent VOIP packets further comprises:

storing the dynamically negotiated VOIP port;

filtering all packets coming through the network switch having the dynamically negotiated VOIP port associated therewith; and

classifying filtered packets in accordance with predefined filtering actions.

11. A method for switching VOIP packets in a data network as recited in claim 10, wherein the step of storing the dynamically negotiated VOIP port further comprises generating a filter corresponding to the dynamically negotiated VOIP port and storing the generated filter in a filter table associated with a fast filtering processor.

12. A method for switching VOIP packets in a data network as recited in claim 10, wherein the filtering step further comprises the steps of:

applying a filter mask to a packet header; and

comparing unmasked header information to an entry in a filter able; and

determining a match between the unmasked header information and the entry in the filter table.

13. A method for switching VOIP packets in a data network as recited in claim 10, wherein the step of classifying filtered packets further comprises taking a filtering action upon a filtered packet in accordance with predetermined actions

stored in a filter action table.

14. A method for switching VOIP packets in a data network as recited in claim 13, wherein the filtering action comprises at least one of modifying a priority associated with the filtered packet, modifying a differentiated services parameter of the filtered packet, modifying a type of service parameter of the filtered packet, sending the filtered packet to a CPU, and dropping the filtered packet.

15. A method for switching VOIP packets, said method comprising the steps of:

filtering packets received in a network switch to trap at least one VOIP call setup message;

determining a dynamically negotiated VOIP port;

filtering all subsequent packets associated with the dynamically negotiated VOIP port; and

taking predefined filtering actions upon the subsequent packets.

16. A method for switching VOIP packets as recited in claim 15, wherein the step of filtering packets to trap at least one VOIP call setup message further comprises the step of filtering packets with a fast filtering processor to determine if a packet header contains a predefined well known port therein.

17. A method for switching VOIP packets as recited in claim 15, wherein the step of determining a dynamically negotiated VOIP port further comprises the steps of:

transmitting packets from a capabilities exchange protocol message to a CPU;

decoding the capabilities exchange protocol message to determine the dynamically negotiated VOIP port; and

storing a filter corresponding to the dynamically negotiated VOIP port in a fast filtering processor.

18. A method for switching VOIP packets as recited in claim 15, wherein the step of filtering all subsequent packets associated with the dynamically negotiated VOIP port further comprises the steps of:

applying a filter to all packets being switched through the network switch to determine which packets are associated with the dynamically negotiated VOIP port; and

applying a filtering action to all packets determined to be associated with the dynamically negotiated VOIP port,

wherein the filtering action includes modifying a priority of a packet in order to reduce network transmission delay for the packet.

19. A method for switching VOIP packets as recited in claim 18, wherein the step of modifying the priority includes at least one modifying a priority associated with the packet, modifying a differentiated services parameter of the packet, modifying a type of service parameter of the packet, sending the packet to a CPU, and dropping the packet.

20. A network switch for switching VOIP packets, said network switch comprising:

at least one data port interface controller supporting a plurality of data ports for transmitting and receiving data;

a fast filtering processor in communication with the at least one data port interface; and

at least one filtering table in communication with the fast filtering processor,

wherein the fast filtering processor is configured to snoop packets being transmitted through the network switch to trap a VOIP call setup message, and thereafter, determine a dynamically negotiated VOIP port so that all subsequent VOIP packets can be filtered and assigned an appropriate priority.

21. A network switch as recited in claim 20, wherein said fast filtering processor further comprises:

a filter unit for constructing and applying a filter to selected fields of an incoming packet, said filter unit including filter logic for selecting desired fields of the incoming packet and copying selected field information therefrom, said filtering logic constructing a field value based upon the selected fields, wherein the filter logic applies a plurality stored field masks on the field value; and

a rules table containing a plurality of rules entries,

wherein the filter logic performs a lookup of the rules table in order to determine actions to be taken based upon the result of a comparison between the field value and the stored filter masks and the rules table lookup.

22. A network switch as recited in claim 21, wherein the filter logic is configured to perform a binary search of the rules table in order to determine a match.

23. A network switch as recited in claim 21, wherein said network switch includes a CPU interface, and wherein the rules table is programmable by a remote CPU through the CPU interface.

24. A network switch as recited in claim 21, wherein the filter unit can be configured to modify incoming packets to change a priority handling field therein.

25. A network switch as recited in claim 21, wherein the rules table, the filter unit, and the CPU interface are implemented on a single silicon substrate.

26. A network switch as recited in claim 21, wherein said filter logic copies the selected field information from a plurality of fields of an incoming packet, and constructs a field value of a predetermined size based upon the selected field information.

27. A network switch as recited in claim 20, said network switch further comprising:

a memory management unit in communication with said at least one data port interface controller;

a memory interface in communication with said at least one data port interface controller, wherein said memory interface is configured to communicate with a memory; and

a communication channel, said communication channel for communicating data and messaging information between said at least one data port interface controller, said memory interface, and said memory management unit,

wherein said memory management unit is configured to route data received from said at least one data port interface controller to said memory interface.

28. A network switch as recited in claim 27, wherein said memory interface further comprises:

an internal memory; and

an external memory interface for communicating with an external memory.

29. A network switch as recited in claim 20, wherein said fast filtering processor is programmable by inputs from a CPU through a CPU interface.

30. A network switch as recited in claim 23, wherein said fast filtering processor filters the packets independent of the CPU interface.

add B'